II. AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions, and listings, of claims in the application:

1. (Original) A tree walking system, comprising:

a binding system for binding a tree observer with a tree, for binding node patterns to node observers to generate at least one node pairing, and for binding the tree observer to at least one node pattern-node observer pairing;

tree walking logic for systematically walking through nodes within the tree;

a pattern testing system for determining if an encountered node matches one of the node patterns;

an event manager for generating an encountered event when one of the node observers is bound to a matching node pattern; and

a pruning system that can deactivate the tree observer with respect to sub-nodes of the encountered node if a bound node observer determines that there is no interest in the sub-nodes.

2. (Original) The tree walking system of claim 1, wherein the encountered event is handled by the bound node observer.

3. (Original) The tree walking system of claim 1, wherein the tree walking logic walks through the tree in a top down hierarchal manner.

4. (Original) The tree walking system of claim 1, wherein the pruning system can reactivate a deactivated tree observer after the sub-nodes of the encountered node have been walked.

5. (Original) The tree walking system of claim 1, wherein the event manager generates a completed event for each node observer that received an encountered event and that did not cause the tree observer to become deactivated.

6. (Original) The tree walking system of claim 5, wherein the completed event can cause the tree walking logic to repeat the walk through the sub-nodes.

7. (Original) The tree walking system of claim 1, wherein the pruning system can further cause the tree walking logic to bypass walking of the sub-nodes if the tree observer has been deactivated and no other active tree observers exist.

8. (Original) A system for analyzing a tree of hierarchical data, comprising:

a system for binding a plurality of tree observers to a tree, wherein each tree observer is further bound to a set of node patterns and a set of node observers;

tree walking logic for systematically walking through nodes within the tree;

a first pruning system that can be instructed by a node observer bound with an associated tree observer to deactivate the associated tree observer until a set of sub-nodes for the encountered node has been walked; and a second pruning system that can instruct the tree walking logic not to walk the set of sub-nodes for the encountered node.

9. (Original) The system of claim 8, wherein the second pruning system will cause the set of sub-nodes not to be walked only if all of the plurality of tree observers have been deactivated.

10. (Original) The system of claim 8, further comprising a pattern testing system for determining if the encountered node matches one of the node patterns.

11. (Original) The system of claim 8, further comprising an event manager for generating an encountered event when one of the node observers is bound to a matching node pattern.

12. (Currently Amended) A <u>computer implemented</u> method for analyzing a tree of hierarchical data, comprising the steps of:

binding a plurality of tree observers to a tree, wherein each tree observer is further bound to a set of node patterns and a set of node observers;

systematically walking through nodes within the tree;

generating an encounter event and handling the encounter event with a bound node observer when one of the node patterns matches an encountered node;

deactivating the tree observer associated with the bound node observer if the bound node observer determines that a set of sub-nodes of the encountered node should be pruned; and bypassing the walking of the set of sub-nodes if all of the plurality of tree observers have been deactivated.

13. (Original) The method of claim 12, comprising the further step of generating a completed event for each node observer that received an encountered event and that did not cause the tree observer to become deactivated.

14. (Original) The method of claim 12, comprising the further step of reactivating the tree observer associated with the bound node observer after the set of sub-nodes of the encountered node have been walked.

15. (Original) The method of claim 12, comprising the further step of reactivating the tree observer associated with the bound node observer after set of sub-nodes of the encountered node have been bypassed.

16. (Original) The method of claim 12, comprising the further step of walking the sub-nodes if at least one tree observer is active.

17. (Original) A program product stored on a recordable medium, which when executed, analyzes a tree of hierarchical data, the program product comprising:

program code configured to bind a plurality of tree observers to a tree, wherein each tree observer is further bound to a set of node patterns and a set of node observers;

program code configured to provide tree walking logic for systematically walking through nodes within the tree;

program code configured to provide a first pruning system that can be instructed by a node observer bound with an associated tree observer to deactivate the associated tree observer until a set of sub-nodes for an encountered node has been walked; and

program code configured to provide a second pruning system that can instruct the tree walking logic not to walk the set of sub-nodes for the encountered node.

18. (Original) The program product claim 17, wherein the second pruning system will cause the set of sub-nodes not to be walked only if all of the plurality of tree observers have been deactivated.

19. (Original) The program product claim 17, further comprising program code configured to provide a pattern testing system for determining if the encountered node matches one of the node patterns.

20. (Original) The program product claim 17, further comprising program code configured to provide an event manager for generating an encountered event when one of the node observers is bound to a matching node pattern.